

## **АРХИТЕКТУРА СОВРЕМЕННОГО WEB-ПРИЛОЖЕНИЯ НА ПРИМЕРЕ ПРОЕКТА «ВЕБ-САЙТ КНИЖНОГО КЛУБА»**

Листопадов С.А.

*Институт информационных технологий БГУИР, г. Минск, Республика Беларусь*  
[s.listopadov@bsuir.by](mailto:s.listopadov@bsuir.by)

Бильдь А.Е.

*УО «Национальный детский технопарк», г. Минск, Республика Беларусь*  
[nastyabild1@gmail.com](mailto:nastyabild1@gmail.com)

Данная статья описывает общие проблемы при разработке архитектуры приложения учащейся УО «Национальный детский технопарк». Кроме того, в статье приводится назначение и описание основных модулей проекта. Для упрощения понимания информации, для используемых в статье терминов и понятий приводится определение.

Ключевые слова: веб-приложение; программирование; php; KnockoutJS; MariaDB; веб-разработка; сайт.

Главной задачей преподавателя при работе по направлению «Web-дизайн» является обучение технологиям, которые используются в профессиональной веб-разработке. Простое заучивание названий функций, методов, синтаксиса языка и алгоритмов действий не может научить решать реальные задачи. Ученик сможет повторить, но он не будет понимать, зачем применяются те или иные технологии и алгоритмы.

Оптимальным является путь, в ходе которого по мере построения приложения, увеличения количества компонентов и возрастания сложность задач, ученика направляют, подсказывают опыт других разработчиков, помогают правильно сформулировать описание проблемы или, например, предлагают рассмотреть архитектуру фреймворков и библиотек и т.д.

Именно такой способ был использован при подготовке проекта по теме «Книжный клуб» во время обучения в УО «Национальный детский технопарк». Реализация проекта предполагала создание веб-сайта, посвящённого художественной литературе и потребовала получения дополнительных навыков, а также освоения таких технологий веб-разработки, как PHP, SQL, JavaScript и др. Поскольку предстояло хранить большой объём информации, возник вопрос об использовании базы данных (далее – БД).

*Примечание:*

PHP (Hypertext Preprocessor) – серверный язык программирования для веб-разработки [1].  
SQL (Structured Query Language) – язык программирования для управления базами данных [2].

JavaScript – язык программирования, который обычно используется как встраиваемый инструмент для программного доступа к различным объектам приложений [3].

База данных – организованная структура, которая предназначена для хранения, изменения и обработки данных.

Самой большой проблемой являлось соединение всех компонентов проекта в единое целое. Проект разрастался, появилась необходимость дополнительной проработки архитектуры веб-приложения. Чтобы лучше погрузиться в проблемы разработки подобного приложения, необходимо представлять структуру данных (рисунок 1).

Каждая книга включает в себя основную информацию: краткое описание, автор, дата написания, жанр. Книга может быть написана несколькими авторами и относиться к нескольким жанрам и поджанрам. Также в БД присутствует таблица пользователей веб-ресурса и их иерархия.

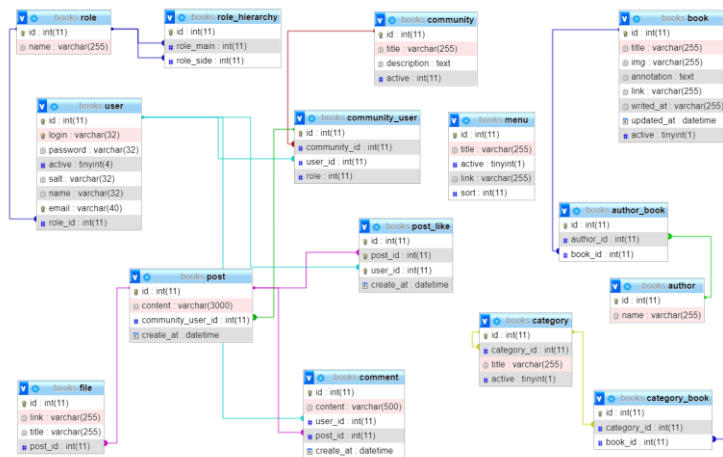


Рисунок 1 – Схема базы данных

Для работы с БД использовался язык программирования SQL. Логика построения веб-страницы крайне сложна для статичной вёрстки. Веб-страницы динамичны, отзывчивы на действия пользователей. В связи с этим было принято решение использовать библиотеку KnockoutJS. Это намного облегчило задачу формирования страницы.

*Примечание:*

KnockoutJS – библиотека на языке JavaScript, которая использует шаблон MVVC (Model-View-View-Model) для создания динамических пользовательских интерфейсов [4].

Для того чтобы снять нагрузку с сервера, использовалась технология AJAX. Это позволило приложению обрабатывать запросы к серверу в фоновом режиме и не перезагружать каждый раз страницу. При этом с сервера загружались только нужные данные.

*Примечание:*

AJAX (Asynchronous Javascript and XML) – технология, которая позволяет обрабатывать запросы к серверу без перезагрузки страницы [5].

Исходя из реалий проекта, в основе веб-приложения использовался паттерн проектирования MVC [6]. Структура приложения была разделена на три части, каждая из которых отвечает за свои задачи:

Model (модель) – хранилище данных, используемых во всём приложении. Она должна обеспечивать доступ к данным для их просмотра, отбора или записи;

View (представление) – отвечает за отображение данных пользователю;

Controller (контроллер) – состоит из действий, которые отвечают за обработку запроса пользователя.

Рассмотрим точку входа приложения. Вначале URL передаётся в маршрутизатор, который возвращает ассоциативный массив, хранящий в себе название контроллера и действие, которое отвечает за обработку данного запроса. Далее идёт подключение к БД. И, наконец, создаётся экземпляр класса нужного контроллера и вызывается действие.

*Примечание:*

URL (Uniform Resource Locator – унифицированный указатель ресурса) – это форма уникального адреса конкретного веб-ресурса в сети Интернет.

Ознакомимся ближе с компонентами приложения. Начнём с маршрутизатора. Сервис состоит из базы маршрутов и методов поиска. В качестве параметров сервис принимает URL. Далее осуществляется поиск в базе маршрутов, и в случае совпадения возвращается информация о нужном действии.

Для упрощения работы с представлениями был создан сервис Template. Данный сервис помогает генерировать HTML вёрстку. Он поддерживает многоуровневый рендеринг, что даёт возможность использовать шаблоны внутри других шаблонов. Принцип работы шаблонизатора заключается в том, что у нас есть некоторая HTML-разметка, которой необходимы данные для заполнения. Вёрстка и данные заключаются в «буфер», что позволяет при необходимости вызвать сервис создания шаблона неограниченное количество раз.

Сервис MariaDBORM был разработан для облегчения работы с БД. Данный сервис не только позволяет подключаться к БД, но и помогает конструировать запросы. Каждая команда SQL формируется вызовом метода класса нашего сервиса. При вызове метода к формируемому запросу добавляется необходимая команда и аргументы. Сервис поддерживает вложенные запросы для различных команд, таких как WHERE, JOIN и другие.

Отдельного упоминания требует модуль ServiceContainer. Его суть в том, чтобы сделать приложение менее зависимым от конкретных сервисов и упростить использование компонентов. Создается экземпляр сервиса и передаётся в сервис-контейнер. В результате можно использовать нужные сервисы множество раз из любой точки веб-приложения.

Для контроля версий приложения использован GIT. Он позволяет упорядоченно хранить код на удалённом сервере. Это даёт возможность при необходимости «откатиться» на прошлую версию проекта.

*Примечание:*

GIT – распределённая система управления версиями [7].

В результате работы над проектом обучающейся были выполнены поставленные задачи: изучены основы языков программирования PHP, SQL, JavaScript; разработана архитектура веб-приложения (за основу был взят паттерн программирования MVC); созданы необходимые для работы сервисы, такие как маршрутизатор, шаблонизатор, MariaDBORM и ServiceContainer.

## Литература

1. PHP: Hypertext Preprocessor [Электронный ресурс]. – Режим доступа: <https://www.php.net/>. – Дата доступа: 08.11.2021.
2. Что такое SQL – Senior.ua [Электронный ресурс]. – Режим доступа: <https://senior.ua/articles/chto-takoe-sql-vse-chto-vam-nuzhno-znat-o-yazyke-strukturirovannyh-zaprosov>. – Дата доступа: 08.11.2021.
3. Введение в JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/intro>. – Дата доступа: 08.11.2021.
4. Knockout.js [Электронный ресурс]. – Режим доступа: <https://knockoutjs.com/>. – Дата доступа: 08.11.2021.
5. AJAX – Руководства Web-разработчика | MDN [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/Guide/AJAX>. – Дата доступа: 08.11.2021.
6. MVC – модель-представление-контроллер – Веб Креатор [Электронный ресурс]. – Режим доступа: <https://web-creator.ru/articles/mvc>. – Дата доступа: 08.11.2021.
7. Git [Электронный ресурс]. – Режим доступа: <https://git-scm.com/>. – Дата доступа: 08.11.2021.

## **APPLICATION ARCHITECTURE ON THE EXAMPLE OF THE PROJECT "WEBSITE FOR A BOOK CLUB"**

Listopadov S.A.

*Institute of information technologies BSUIR, Minsk, Republic of Belarus*

Bil'd' A.E.

*National Children's Technopark, Minsk, Republic of Belarus*

This article describes general problems which appeared during the development of the application architecture by the student of the EE “National Children Technopark”. Moreover, the article provides the purpose and description of the main modules of the project. The article also provides definitions for the terms and concepts used in it to make understanding easier.

Keywords: web-application; programming; php, KnockoutJS; MariaDB; web-development; web-site.